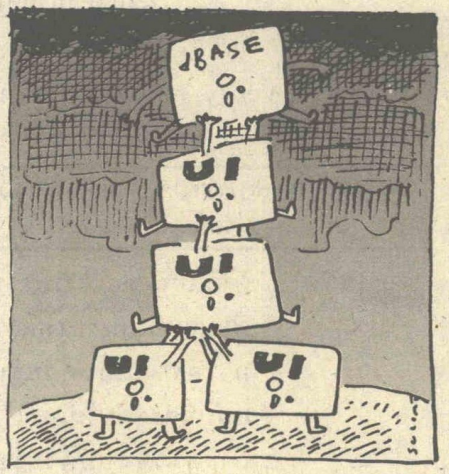
มุมนักเล่น

dBASE

ยืน ภู่วรวรรณ

โปรแกรม UI
เพื่อการสร้างโปรแกรม



สร้างสรรค์โปรแกรม บนดีเบสได้อย่างจุใจ ใช้งานง่ายและรวดเร็ว

ดีเบสเป็นโปรแกรมที่แพร่หลายไปทั้งโลก ผู้ใช้พีซีเกือบทุกคนเคยได้เรียนรู้หรือใช้งานกันมาแล้ว ดีเบสเป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่มีประสิทธิภาพในการจัดการข้อมูล ผู้เขียนโปรแกรมด้วยดีเบสบ่อย ๆ สามารถกำหนดให้ระบบงานตรงตามความต้องการของผู้ใช้ได้ ดีเบสจึงเป็นโปรแกรมยอดนิยมในเรื่องการจัดการฐานข้อมูลบนพีซี ดังนั้นจึงมีโปรแกรมที่ทำงานโดยใช้กฎไวยากรณ์เดียวกับดีเบสหลายโปรแกรม เช่น ฟอกซ์เบส คลิปเปอร์ ฯลฯ

เพื่อสนับสนุนการทำงานของดีเบสให้ดียิ่งขึ้น จึงมีผู้พัฒนาโปรแกรมสนับสนุนอื่น ๆ อีกหลายตัวในลักษณะที่แตกต่างกัน เช่น การสร้างกราฟ การแสดงผลข้อมูล หรือแก้ไขข้อมูล ยูทิลิตี้ประกอบ ตลอดจนโปรแกรมสนับสนุนงานเฉพาะเรื่องอีกมากมาย

UI เป็นโปรแกรมที่พัฒนาขึ้นมาเพื่อใช้ช่วยการเขียนโปรแกรมดีเบส โดยผู้ใช้ไม่จำเป็นต้องมีความรู้ทางด้านการเขียนโปรแกรมมากนักก็สามารถสร้างระบบงานที่ซับซ้อนได้ UI เป็นโปรแกรมที่สร้างเอาต์พุตเป็นตัวซอร์สโค้ดของดีเบสทรีพลัส ดีเบสโฟร์ คลิปเปอร์ ฟอกซ์เบส หรือแม้แต่สร้างเอาต์พุตออกมาเป็นโปรแกรมที่เขียนด้วยภาษาซี

การใช้งาน UI จึงเหมาะกับผู้พัฒนาโปรแกรมบนดีเบสที่ต้องการเครื่องมือช่วย เพื่อทำให้เขียนโปรแกรมได้ง่ายและรวดเร็ว UI สร้างโปรแกรม PRG ที่สามารถนำมารวมหรือร่วมทำงานกับโปรแกรมอื่นได้ การเขียนโปรแกรมด้วย UI ทำได้รวดเร็วและประหยัดเวลา

เริ่มต้นใช้ UI

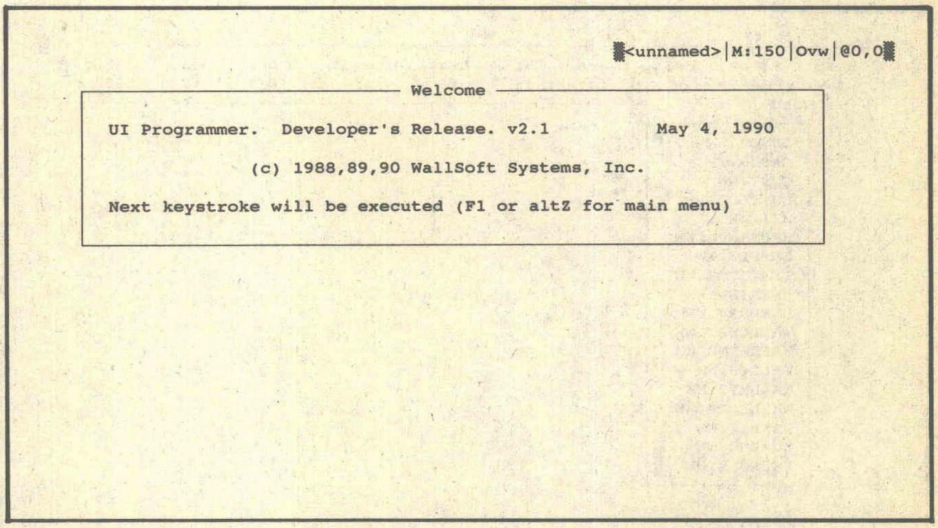
UI เป็นโปรแกรมที่พัฒนาโดยบริษัท Wallsoft System Inc. UI ที่ผู้เขียนแนะนำนี้เป็นโปรแกรมเวอร์ชัน 2.1 ที่พัฒนาออกใช้ในปี ค.ศ. 1990 ประกอบด้วยแผ่นดิสค์ 4 แผ่น (ขนาด 360 กิโลไบต์) แต่เมื่อติดตั้งลงในฮาร์ดดิสค์แล้วจะแยกไดเรคทอรีย่อยลงไปโดยใช้ชื่อไดเรคทอรี TEM สำหรับเก็บเทมเพลต TLB เก็บแฟ้มที่เป็นไลบรารี GEN เก็บแฟ้มที่ได้มาจากการสร้างเป็นซอร์สโค้ด ส่วน DIC เก็บรายละเอียดดาต้าดิกชันนารี ไดเรคทอรี

ภาพประกอบ : สุชาติ สหชาติโภคานันท์

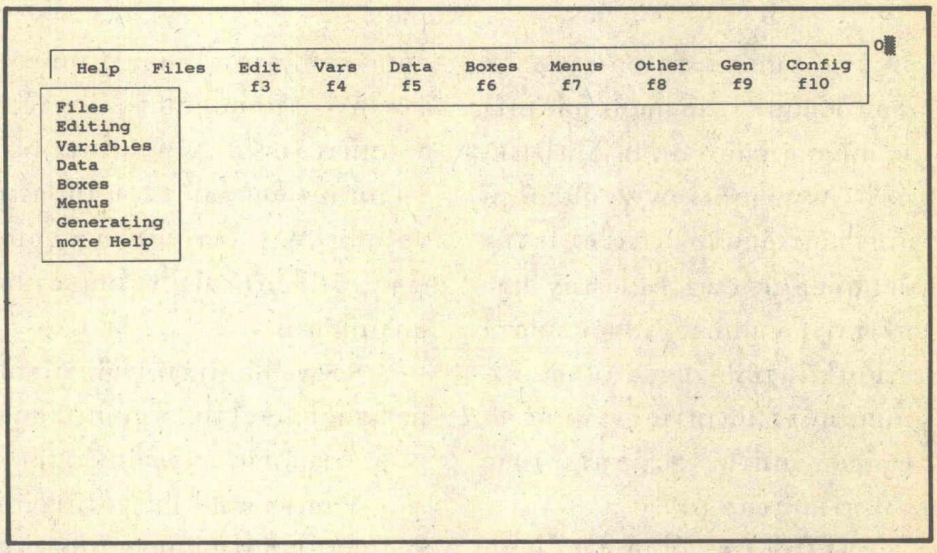
ย่อย WW เก็บไฟล์ที่ออกแบบก่อนการนำมาสร้างเป็นซอร์สโค้ด DBF เก็บไฟล์ประเภท .dbf และ .NDX เก็บไฟล์ประกอบอินเด็กซ์ไฟล์

เมื่อติดตั้งเสร็จแล้วจะได้ UI.EXE เป็นโปรแกรมหลักที่จะเรียกใช้โปรแกรม UI.EXE นี้เป็นโปรแกรมที่ใช้สร้างระบบงานอื่นที่ต้องการ การเรียกใช้จะเรียก UI หลังจากนั้นรอสักครู่จะปรากฏเมนูในลักษณะโลโก้ที่แสดงผลดังรูปที่ 1

เมื่อกดคีย์ใด ๆ จะพบเอดิเตอร์ที่จะให้เขียนระบบงานที่ต้องการโดยแสดงเป็นหน้ากระดาษว่าง ๆ หนึ่งหน้า หน้ากระดาษนี้มุมขวาจะมี *<unnamed> หมายถึงยังไม่มีการตั้งชื่อ M:150 หมายถึงหน่วยความจำที่เหลืออีก 150 กิโลไบต์ ถัดจากนั้น Ovw หมายถึงการเขียนทับ @0,1 หมายถึงตำแหน่งเคอร์เซอร์ที่อยู่บนกระดานจอภาพ



รูปที่ 1 เมนูในลักษณะโลโก้ของโปรแกรม UI



รูปที่ 2 เมนูขอความช่วยเหลือแบบพูลดาวน์

เมนูของ UI

เมื่อกดแป้น F1 หลังจากอยู่ที่เอดิเตอร์ของ UI จะพบเมนูแสดงผลขึ้นมาตรงตอนบนประกอบด้วย

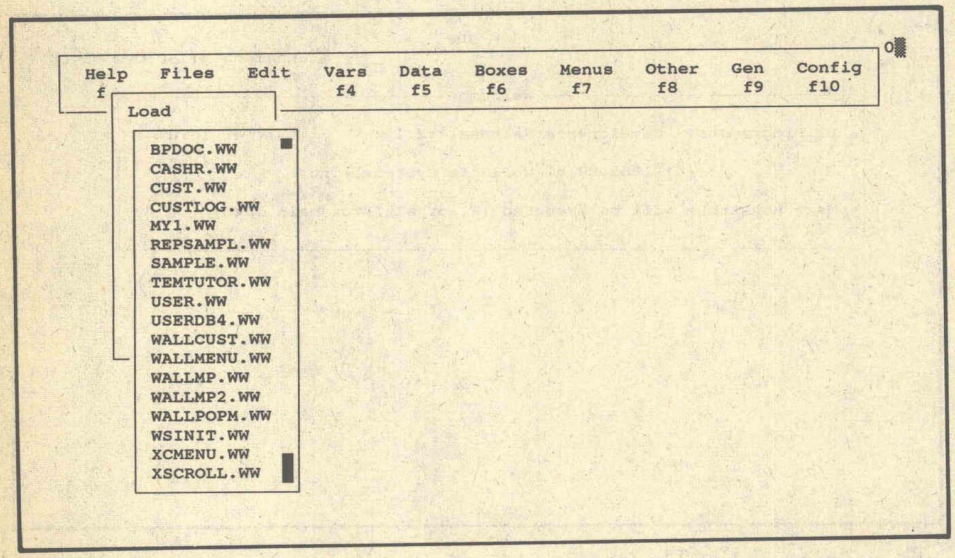
Help (F1) หมายถึงการขอความช่วยเหลือ ซึ่งหากกดแป้นรีเทอร์นจะปรากฏพูลดาวน์เมนู ลักษณะของเมนูเป็นดังรูปที่ 2

การขอความช่วยเหลือนี้เราเลือกขอคำอธิบายได้ เช่น เมื่อแถบสว่างอยู่ที่ Files กดรีเทอร์นจะได้รายละเอียดข้อความคำอธิบาย ซึ่งให้รายละเอียดของ UI ในกลุ่มคำสั่งไฟล์ ในการขอความช่วยเหลือของ UI ยังให้รายละเอียดไม่มากพอ ส่วนใหญ่จะเป็นคำอธิบายอย่างกว้าง ๆ ว่าเมนูข้อนี้ทำอะไรได้บ้าง ผู้ใช้จะต้องดูจากคู่มือเป็นหลัก คู่มือของ UI อาจจะเล่มโตสักหน่อย แต่เขียนแบบลักษณะรวบรวมเนื้อหาสำคัญจึงอ่านยาก หนังสือคู่มือดังกล่าวจึงไม่ใช่คู่มือผู้ใช้ แต่จะเป็นคู่มือรวมคำสั่งหรือข้อมูลของ UI

เมื่อต้องการเลิกจากการขอความช่วยเหลือให้กดแป้น Esc

Files เมนูหรือ F2 เป็นกลุ่มเมนูสำหรับการโหลดชิ้นงานที่เป็นรูปฟอร์มและการเรียกรูปแบบสกรีนออกมาใช้ โดยรูปแบบเมนูจะเป็นดังรูปที่ 3

คำสั่ง Load คือการโหลดชิ้นงานที่ได้รับการออกแบบไว้แล้ว โดยชิ้นงานจะอยู่ในแฟ้มที่มีการขยายเป็น WW แฟ้ม .WW นี้ประกอบด้วยรายละเอียดที่ UI รู้จักดี เช่น กรอบสี่เหลี่ยม ฟิลด์-ข้อมูล ข้อความ หรือข้อกำหนดเกี่ยวกับเมนูคำสั่งต่าง ๆ การเรียกแฟ้ม .WW นี้ทำได้ด้วยการเลือกแถบสว่าง แล้วกด



รูปที่ 3 รูปแบบเมนูเมื่อเลือกคำสั่ง Files

รีเทอร์น จากนั้นจะปรากฏ .WW ให้เลือก ใช้แถบสว่างเลือกแฟ้มที่ต้องการได้ แฟ้มเหล่านี้เก็บรวบรวมไว้ในไดเรคทอรี .WW แฟ้ม .WW เป็นแฟ้มที่เก็บรายละเอียดของการเชื่อมโยงทั้งข้อมูลที่อยู่ใน Data Dictionary และสร้างการออกแบบจอภาพข้อมูลเหล่านี้จะเก็บไว้ในรูปฟอร์มของ UI เอง ดังนั้นแฟ้มนี้จึงไม่สามารถที่จะพิมพ์รายละเอียดออกมาได้ จะต้องเรียกโหลดผ่านทางเมนูของ UI

O:Overlay เป็นคำสั่งคล้ายกับคำสั่ง Load แต่หากใช้ Load จะเป็นการโหลดมาบนสกรีนโดยลบของเดิมก่อนแล้วจึงโหลดมาบนสกรีน แต่ Overlay จะโหลด .WW มาทับ .WW เดิมที่อยู่บนจอภาพ จอภาพเดิมจะเป็นแบคกราวนด์ ถ้าหากมีฟิลด์ข้อมูลซ้ำกันจะไม่ได้รับการโหลด ขนาดจำนวนกรอบที่ซ้อนกันมีจำนวนได้ 256 ดังนั้นการโหลด Overlay นี้ช่วยทำให้เราสร้างวินโดว์ซ้อนกันได้

P:Pick to Load เป็นคำสั่งที่เหมือนกับ Load แต่เราสามารถเลือกออบเจ็คต์ที่ต้องการโหลดลงมาบนจอภาพได้ การเลือกออบเจ็คต์นี้จะเป็นส่วนหนึ่งของแฟ้ม .WW หรือของทั้งชิ้นงาน เมื่อเลือกข้อนี้ UI จะให้สกรีนของแฟ้ม .WW โดยให้ผู้ใช้เลือกออบเจ็คต์ เมื่อเลือกได้แล้วก็จะโหลดลงบนสกรีนที่ต้องการ

S:Save เป็นคำสั่งที่ใช้เก็บชิ้นงานที่ออกแบบบนจอไว้ในแฟ้มที่มีนามสกุล .WW การเก็บชิ้นงานจะเก็บทั้งหมด

V:save visible เป็นคำสั่งที่ใช้เก็บชิ้นงานเฉพาะที่ปรากฏบนจอภาพเท่านั้น

C:save chosen เป็นการเลือกออบเจ็คต์บนจอภาพ เพื่อเก็บลงไฟล์ การเก็บของ UI จะเก็บในรูปแบบ .WW เช่นกัน

D:dbase import เป็นคำสั่งการโหลดไฟล์ .PRG ที่มีคำสั่งแสดงผลบนจอภาพ คือคำสั่ง @..SAY UI จะแปลความหมายของคำสั่งแสดงผลบนสกรีนตามรูปแบบของดีเบส คำสั่ง @ นี้รวมไปถึงคำสั่ง BOX ของดีเบสด้วย

A:ascii import เป็นคำสั่งอ่านแฟ้มข้อมูลที่เก็บด้วยเวิร์ดโปรเซสเซอร์เข้ามาบนสกรีนของ UI

X:xport ascii เป็นคำสั่งเก็บรูปแบบจอภาพออกไปเป็นแฟ้มข้อมูลแบบเท็กซ์หรือแอสกี

V:Export video เป็นคำสั่งเก็บภาพที่แสดงผลไปเป็นแฟ้มข้อมูล

หากเลือก Edit เมนู

เมนูข้อนี้มีไว้สำหรับให้แก้ไขรูปแบบที่ออกแบบไว้ ประกอบด้วยคำสั่งหลายคำสั่ง ลักษณะของพูลดาวน์เมนูเมื่อเลือก Edit เป็นดังรูปที่ 4

M:move area เป็นคำสั่งให้ผู้ใช้สร้างบล็อกบนจอภาพเพื่อเลื่อนข้อมูลในบล็อกที่กำหนดไปไว้ที่ใดก็ได้ตามต้องการ การเลื่อนทำได้ง่ายดายเพราะใช้แป้นลูกศรเลื่อนไปในทิศทางที่ต้องการ

C:copy area เป็นคำสั่งคัดลอกกลุ่มข้อความจากที่หนึ่งไปยังอีกที่หนึ่ง การคัดลอกทำคล้ายกับคำสั่ง move

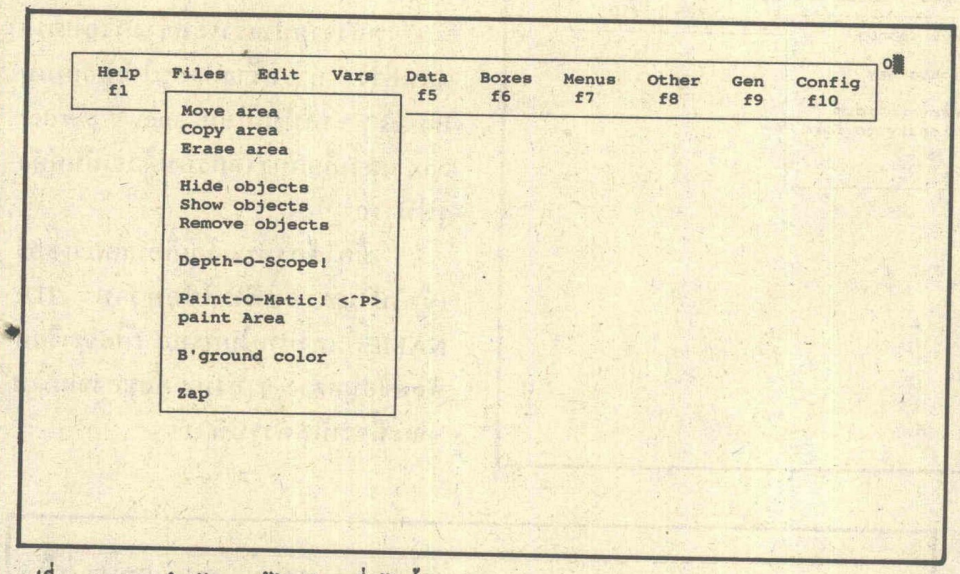
E:erase area เป็นคำสั่งลบกลุ่มข้อความหรือข้อมูลบนสกรีนที่ออกแบบไว้

H:hide object เป็นคำสั่งที่ให้ผู้ใช้นำออบเจ็คต์ที่อยู่บนจอภาพไปซ่อนไว้ไม่ให้แสดงผล เพื่อเป็นประโยชน์ในการแสดงผลแบบมัลติสกรีนหรือให้มีวินโดว์ซ้อนกัน

S:Show object เป็นคำสั่งให้เลือกออบเจ็คต์เพื่อให้แสดงผลบนจอภาพหลังจากที่ซ่อนไว้

R:remove object เป็นคำสั่งสำหรับเลือกออบเจ็คต์หนึ่งที่ต้องการแล้วทำการลบออบเจ็คต์นั้นออกไป

D:depth-O-scope เป็นคำสั่งที่ใช้สำหรับการเลือกกรอบสี่เหลี่ยมที่สร้างขึ้นซ้อนกันหลาย ๆ กรอบให้แสดงผลเป็น



รูปที่ 4 เมนู Edit สำหรับการแก้ไขรูปแบบที่สร้างขึ้น

ลำดับตามตำแหน่งการเลือก

P:paint-O-matic เป็นคำสั่งที่ใช้สำหรับกำหนดสีลงในสกรีนที่ต้องการ

A:paint area เป็นคำสั่งสำหรับระบายสีในกรอบที่กำหนด

B:B'Ground color เป็นคำสั่งเปลี่ยนสีพื้น

Z:zap เป็นคำสั่งลบสกรีนที่ออกแบบมาทั้งหมดให้เป็นสกรีนว่างเพื่อออกแบบใหม่ต่อไป

กลุ่มคำสั่ง Vars

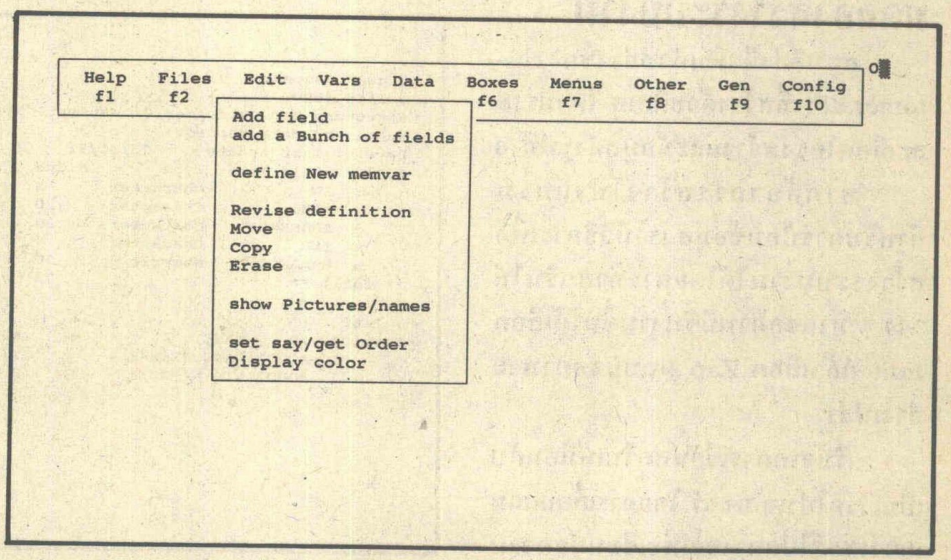
กลุ่มคำสั่งนี้จะเกี่ยวข้องกับการโหลดฟิลด์ของข้อมูลที่ได้รับการกำหนดไว้ในส่วนของ data dictionary เพื่อให้ปรากฏรายละเอียดของฟิลด์ในสกรีนที่ออกแบบ รายละเอียดของเมนูนี้แสดงผลบนจอดังรูปที่ 5

A:add a field เป็นการเลือกฟิลด์ในแฟ้มข้อมูลที่ได้รับการกำหนดไว้ใน data dictionary การเลือกจะเลือกทีละฟิลด์

B:add a Bunch of fields เป็นคำสั่งเลือกกลุ่มของฟิลด์ที่ได้รับการกำหนดไว้ใน data dictionary โดยเลือกมาเป็นกลุ่ม

N:define New Memvar เป็นคำสั่งกำหนดตัวแปรที่จะใช้ในการคำนวณหรือสร้างสูตร

R:revise definition เป็นคำสั่งเปลี่ยนแปลงรายละเอียดของการกำหนดฟิลด์หรือตัวแปร



รูปที่ 5 รายละเอียดของกลุ่มคำสั่ง Vars

move, copy, erase เป็นคำสั่งใช้ในการเลื่อนฟิลด์หรือตำแหน่งตัวแปรที่ต้องการ

คำสั่งที่เหลือเป็นการเลือกกำหนดการแสดงลักษณะ PIC ของตำแหน่งฟิลด์ที่กำหนด การแสดงลำดับการใช้ get และการแสดงผลที่เป็นสี

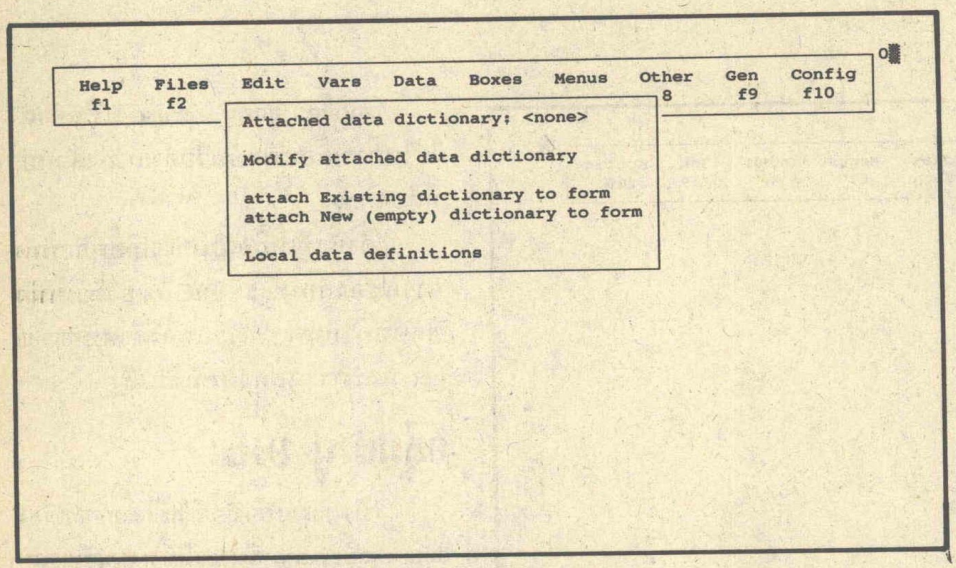
กลุ่มเมนู Data

เป็นกลุ่มคำสั่งที่ใช้ในการสร้าง data dictionary ซึ่งปกติชิ้นงานที่ออกแบบจะต้องมี data dictionary เป็นตัวทำงานร่วม เราสามารถสร้างรายละเอียดข้อมูลส่วนนี้ขึ้นมา เพื่อใช้ต่อเชื่อมกับระบบงานหรือสร้างความสัมพันธ์กันระหว่างแฟ้ม ลักษณะของเมนูแสดงได้ดังรูปที่ 6

กลุ่มคำสั่ง Boxes

เป็นกลุ่มคำสั่งการสร้างกรอบเพื่อใช้งานหรือสร้างวินโดว์ ในการสร้างกรอบทำได้ง่าย เพราะเพียงแต่ใช้แป้นลูกศร เมื่อสร้างกรอบได้แล้วก็สามารถ

```
Help   Files   Edit   Vars   Data   Boxes   Menus   Other   Gen   Config
 f1     f2                                          8      f9    f10

Attached data dictionary: <none>

Modify attached data dictionary

attach Existing dictionary to form
attach New (empty) dictionary to form

Local data definitions
```

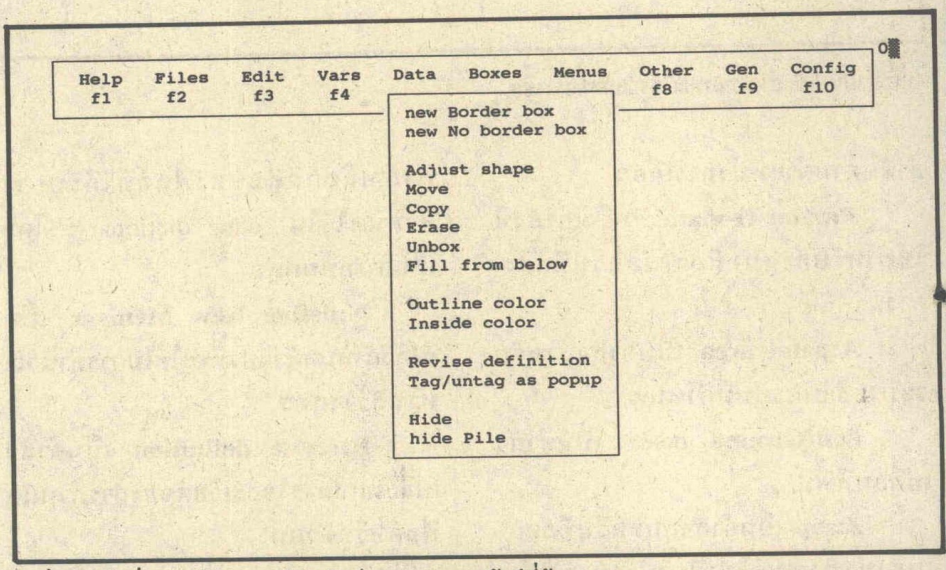


รูปที่ 6 กลุ่มคำสั่งที่ใช้ในการสร้าง data dictionary

หลังจากนั้นเราจะสร้างกรอบตรงตอนล่าง การสร้างกรอบให้เลือกเมนู Boxes จากนั้นเลือก new border box แล้วเลือกการตีกรอบด้วยเส้นคู่ดังรูปที่ 10

เมื่อได้กรอบแล้วเลื่อนเคอร์เซอร์เข้าไปในกรอบพิมพ์ข้อความ ID: NAME:...ลงภายในกรอบ เพื่อจะเรียกฟิลด์ข้อมูลต่าง ๆ ลงมา ผลของจอภาพขณะนี้จะเป็นดังรูปที่ 11

```
Help   Files   Edit   Vars   Data   Boxes   Menus   Other   Gen   Config
 f1     f2     f3     f4                             f8      f9    f10

new Border box
new No border box

Adjust shape
Move
Copy
Erase
Unbox
Fill from below

Outline color
Inside color

Revise definition
Tag/untag as popup

Hide
hide Pile
```



รูปที่ 7 เมนูคำสั่งการสร้างกรอบหรือวินโดว์ ตลอดจนการแต้มสีที่ต้องการ

เติมสีตามที่ต้องการได้ ลักษณะของเมนูมีรายละเอียดดังรูปที่ 7

สำหรับคำสั่งที่เหลือจะอธิบายในโอกาสต่อไป

ทดลองสร้างระบบงาน

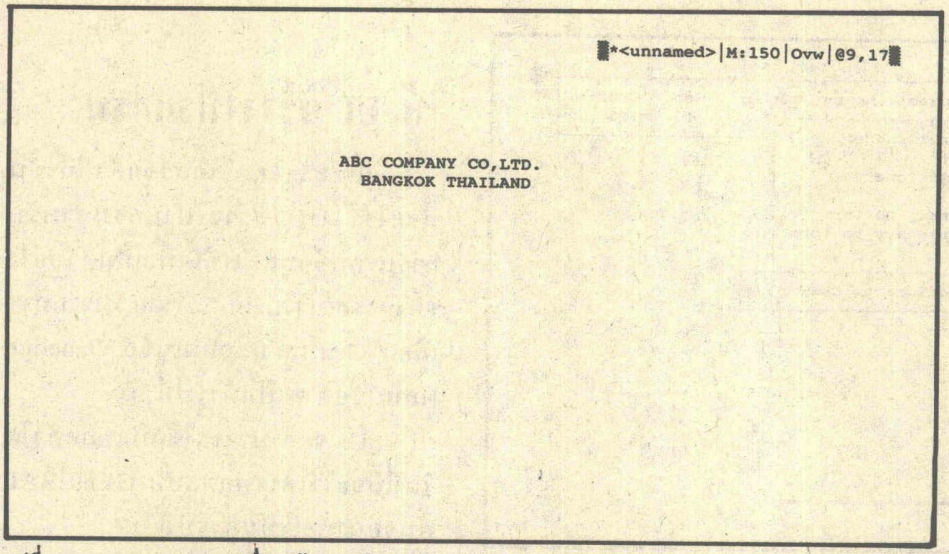
สมมติว่ามีแฟ้มตัวอย่างชื่อ customer.dbf ที่สร้างด้วยดีเบส โดยมีรายละเอียดโครงสร้างและข้อมูลดังรูปที่ 8

หากต้องการสร้างโปรแกรมสำหรับการป้อนข้อมูล เรามีวิธีการเพื่อสร้างระบบงานได้โดยการทำสกรีนให้ว่าง หากจอภาพมีชิ้นงานอื่นให้เลือก Edit แล้วเลือก Zap ผลบนจอภาพจะว่างเปล่า

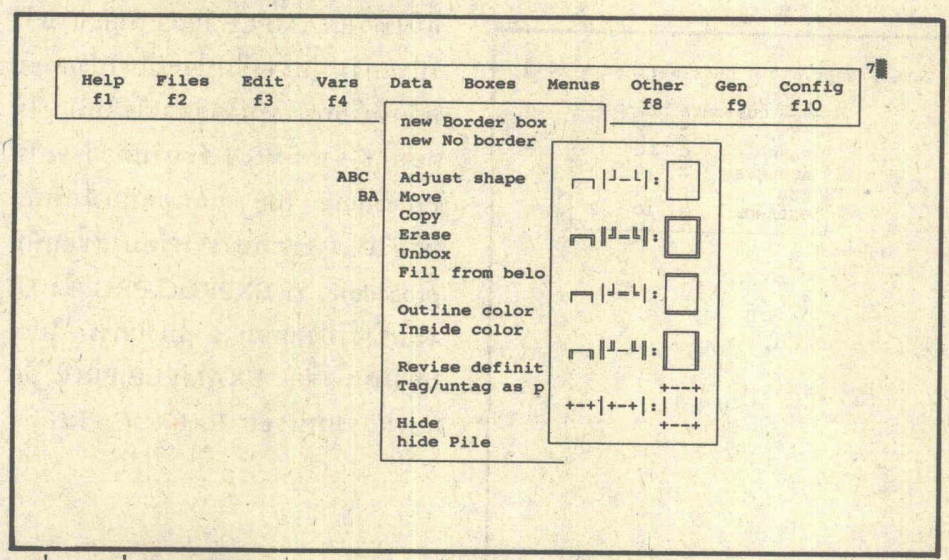
เริ่มจากการเขียนสกรีนเหมือนกับเป็นเวิร์ดโปรเซสเซอร์ โดยการเลื่อนแถบเคอร์เซอร์ไปกลางจอเพื่อเขียนข้อความดังรูปที่ 9

```
.
. use customer
. list stru
Structure for database: C:customer.dbf
Number of data records:        3
Date of last update    : 03/31/91
Field  Field Name  Type        Width    Dec
    1  ID          Character       5
    2  NAME        Character      20
    3  ADDRESS     Character      30
    4  TEL         Character       7
    5  BALANCE     Numeric        10      2
** Total **                       73

. list
Record#  ID     NAME                ADDRESS                        TEL        BALANC
      1  A0001  DANG DEEMAK         1/2 PRAHOLYOTHIN BKK 10900     2347077    25000.0
      2  A0002  NID NARUDEE         2 NEW ROAD BKK 12000           2232234    40000.0
```

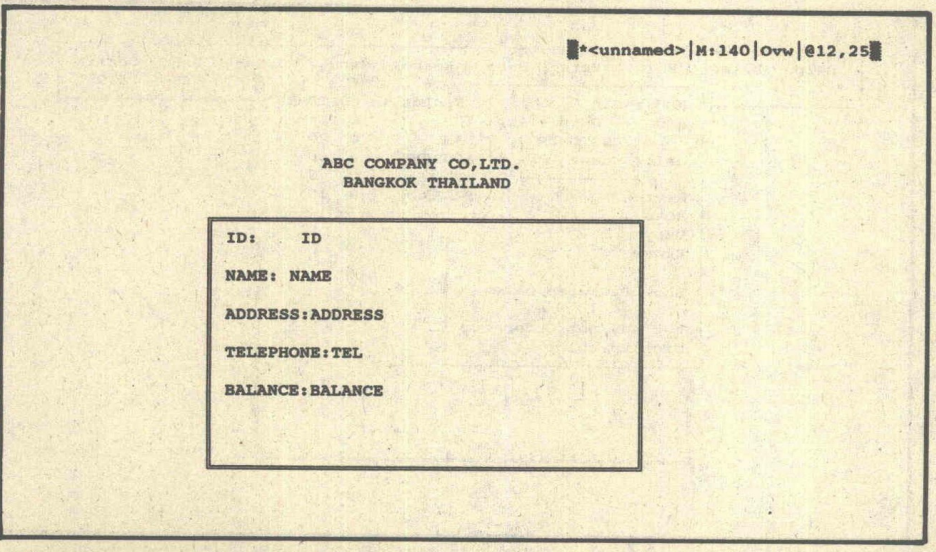
รูปที่ 8 โครงสร้างและข้อมูลในแฟ้มตัวอย่างที่ชื่อ customer.dbf



รูปที่ 9 การแสดงผลบนจอภาพเมื่อเราต้องการเขียนสกรีน



รูปที่ 10 การสั่งตีกรอบด้วยเส้นคู่



รูปที่ 11 ผลบนจอภาพหลังจากเรียกฟิลด์ข้อมูลต่าง ๆ

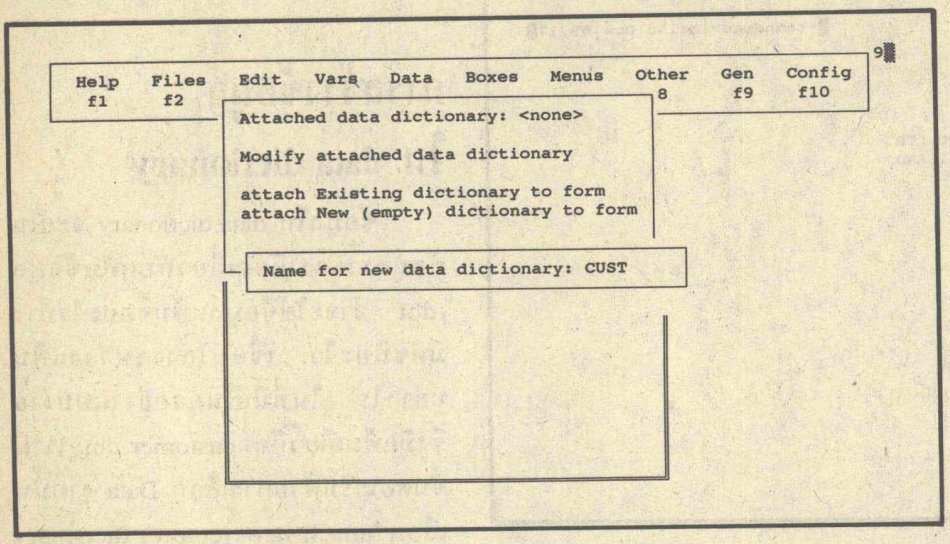
การสร้างข้อมูล ใน data dictionary

ข้อมูลใน data dictionary จะเป็นข้อมูลรายละเอียดเกี่ยวกับแฟ้มข้อมูล .dbf ว่าจะใช้ข้อมูลแฟ้มชื่ออะไรบ้าง มีดัชนีอะไร เชื่อมโยงระหว่างแฟ้มอย่างไร ในที่นี้เป็นตัวอย่างแบบง่าย จึงมีแฟ้มเดียวคือ customer.dbf เรามีขั้นตอนเริ่มด้วยการเลือก Data จากนั้นเลือก attach new (empty) dictionary to form ใส่ชื่อ data dict ที่กำหนดในที่นี้ใช้ชื่อ CUST ผลบนจอภาพเป็นดังรูปที่ 12

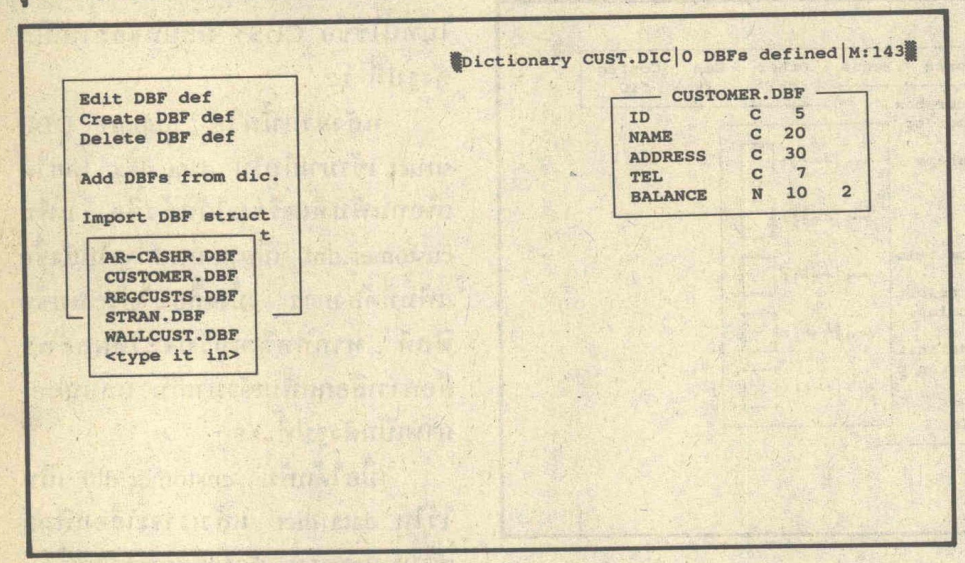
หลังจากนั้นให้ import DBF struct เข้ามาเก็บใน data dict โดยได้เลือกแฟ้มที่เตรียมไว้แล้วคือ แฟ้ม customer.dbf เลื่อนแถบสว่างไปยังชื่อแฟ้มที่ต้องการ เมื่อเลื่อนไปจะพบชื่อฟิลด์ หากกดรีเทอร์นจะได้ผลลัพธ์คือการเลือกแฟ้มเข้ามาเก็บ ผลบนจอภาพเป็นดังรูปที่ 13

เมื่อได้แฟ้ม customer.dbf เก็บไว้ใน data dict แล้วเราจะเลือกฟิลด์เข้ามาบนสกรีน ก่อนเลือกฟิลด์ให้เลื่อนเคอร์เซอร์บนสกรีนไปต่อข้างท้ายคำว่า ID: เลือกเมนู คราวนี้ใช้ Vars แล้วเลือก Add field จะเห็นชื่อ customer ให้กดรีเทอร์น แถบสว่างจะมาอยู่ที่ ID กดรีเทอร์นจะได้ฟิลด์ ID มาแทรกในสกรีน ผลบนจอภาพเป็นดังรูปที่ 14

ทำเช่นเดียวกันโดยเลื่อนเคอร์เซอร์มาต่อท้าย NAME แล้ว add field name ทำจนครบทุกฟิลด์ ผลบนจอภาพจะเป็นดังรูปที่ 15



รูปที่ 12 ผลบนจอภาพเมื่อเราเรียกไฟล์ CUST



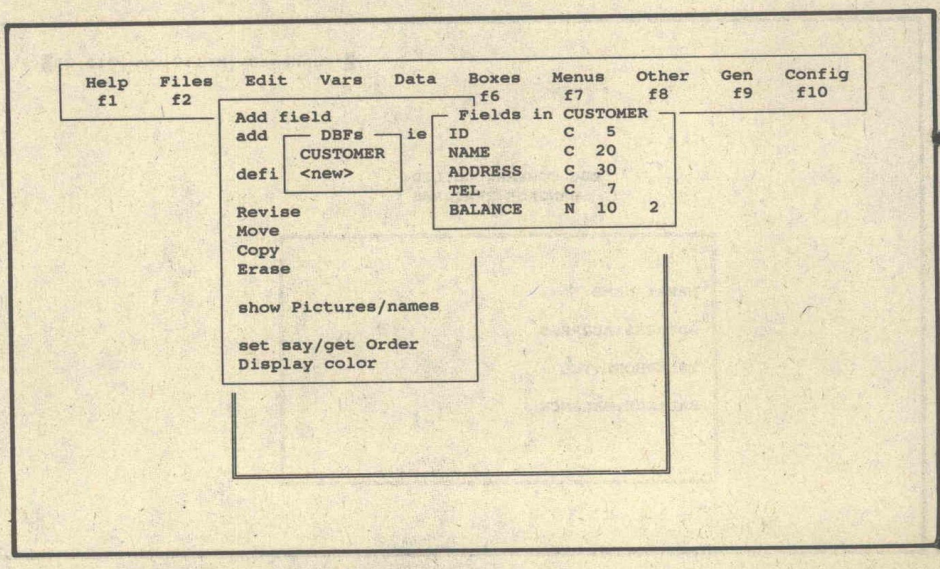
รูปที่ 13 ผลลัพธ์จากการเลือกแฟ้มเข้ามาเก็บ

ให้ UI สร้างโปรแกรม

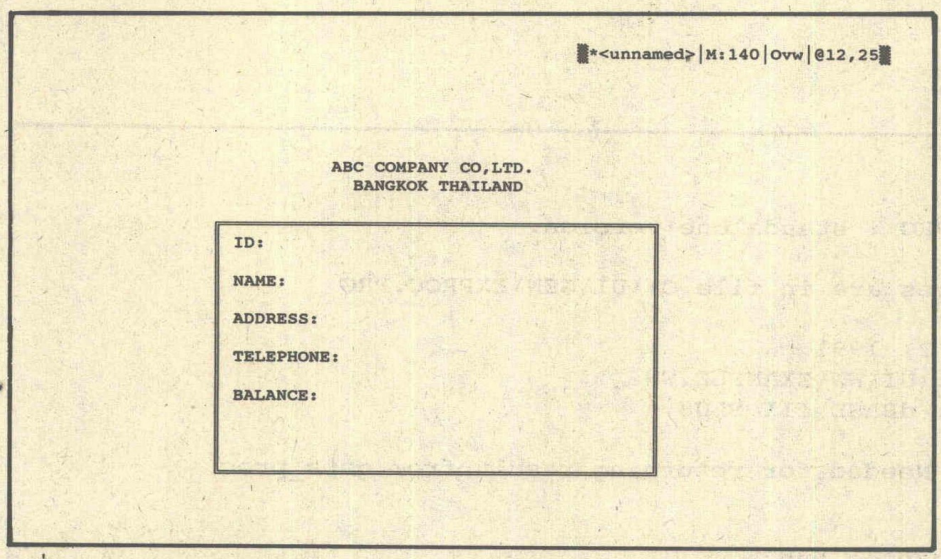
เมื่อสร้างสกรีนเสร็จแล้วก็ถึงขั้นที่จะให้ UI สร้างโปรแกรมสำหรับการกรอกข้อมูล ในที่นี้เพื่อเป็นตัวอย่างของการสร้างโปรแกรม โดยเริ่มจากการเลือก Gen แล้วเลือกเมนูข้อ Generate ผลบนจอภาพเป็นดังรูปที่ 16

UI จะถามว่าจะใช้เทมเพลตตัวใดในที่นี้จะใช้เทมเพลตแบบ GENERAL บนจอภาพจะเป็นดังรูปที่ 17

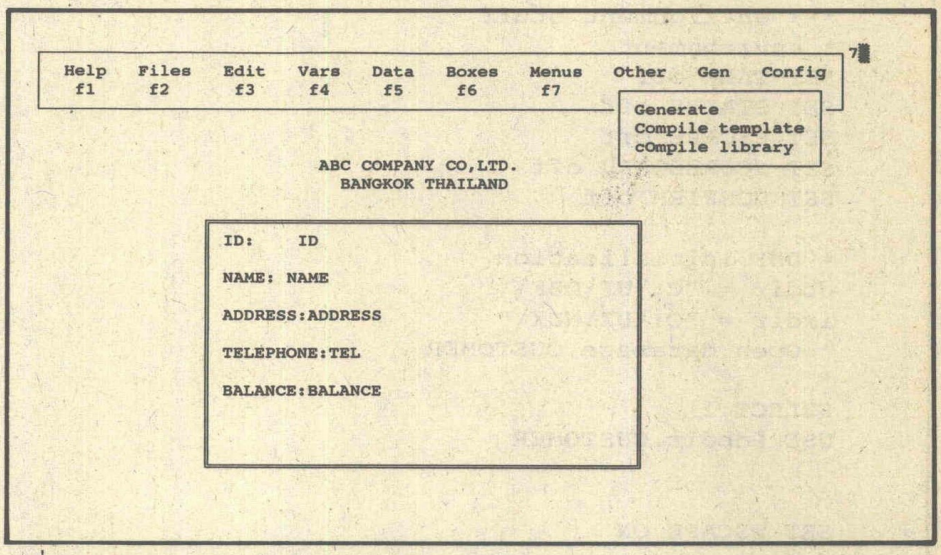
UI จะให้เติมชื่อโปรแกรม ในที่นี้ใช้ EXAMPLE.PRG และถามว่าโปรแกรมที่สร้างขึ้นนี้จะเป็นโปรแกรมหลักที่เรียกทำงานโดยตรงใช่ไหม ให้ตอบ Y จากนั้น UI จะถามต่อว่าจะให้ Procedure file แยกออกแฟ้มหลักหรือไม่ เราจะตอบ Y พร้อมใส่ชื่อแฟ้ม procedure ว่า EXPROC.PRG ซึ่ง UI จะสร้างโปรแกรม 2 โปรแกรม โปรแกรมหนึ่งชื่อ EXAMPLE.PRG อีกโปรแกรมหนึ่งชื่อ EXPROC.PRG



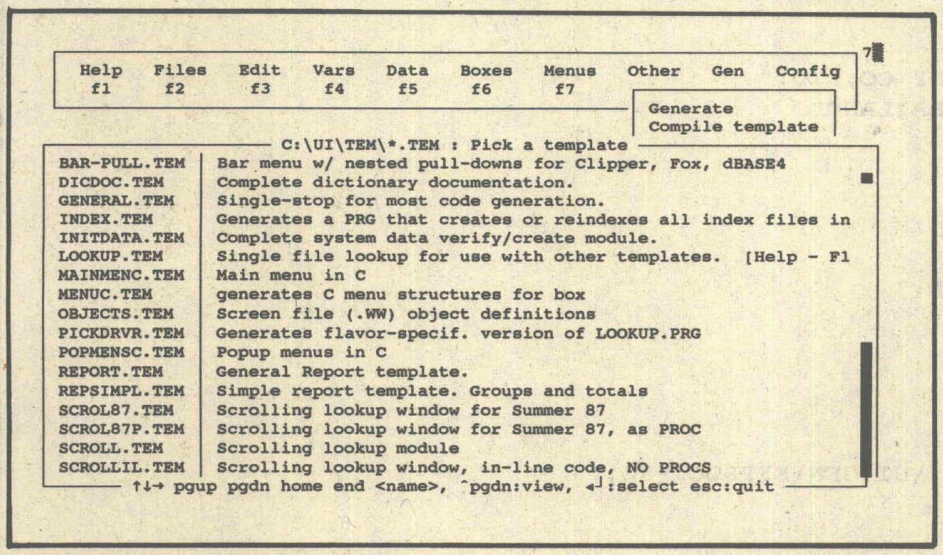
รูปที่ 14 การแสดงผลเมื่อเรียกฟิลด์ ID มาแทรกในสกรีน



รูปที่ 15



รูปที่ 16 การแสดงผลหลังจากเลือกเมนู Generate



รูปที่ 17 การเลือกใช้เทมเพลตในโปรแกรม UI

โปรแกรม EXAMPLE.PRG ที่สร้างขึ้นโดย UI

```
***
*** C:\UI\GEN\EXAMPLE.PRG : standalone version.
*** Purpose: ADD
*** Supporting procedures are in file C:\UI\GEN\EXPROC.PRG
***
*** Generated on April 2, 1991
*** Source .WW file: C:\UI\WW\EXAMPLE.WW
*** Target environment: dBASE III PLUS
***
PUBLIC changed       && Needed for returning result from gets_proc
PRIVATE dbdir, ixdir

SET PROC TO C:\UI\GEN\EXPROC.PRG
*** environment stuff
* environment
SET TALK off
SET STATUS off
SET ESCAPE off
SET SCOREBOARD off
SET CONFIRM off

* DBF initialization
dbdir = "C:\UI\DBF\"
ixdir = "C:\UI\NDX\"
* Open database CUSTOMER
*
SELECT 1
USE &dbdir.CUSTOMER


SET ESCAPE ON

* display fixed text
SET COLOR TO +W/N
CLEAR
@ 6, 28 SAY "ABC COMPANY CO,LTD."
@ 7, 30 SAY "BANGKOK THAILAND"
@ 9,17 TO 22,58 DOUBLE
@ 10,18 CLEAR TO 21,57
@ 10, 19 SAY "ID:"
@ 12, 19 SAY "NAME:"
@ 14, 19 SAY "ADDRESS:"
@ 16, 19 SAY "TELEPHONE:"
@ 18, 19 SAY "BALANCE:"

DO addrec

RETURN
* (Procedures are in C:\UI\GEN\EXPROC.PRG).
```

โปรแกรม EXPROC ที่สร้างขึ้นโดย UI

```
*************************************************************************
*
*               Procedures for EXAMPLE.  File: C:\UI\GEN\EXAMPLE.PRG
*
*************************************************************************
*********************************
* addrec: append new record
*********************************
PROC addrec
PRIVATE currec

*** Create memvar duplicates for all fields
PRIVATE mID, mNAME, mADDRESS, mTEL, mBALANCE

*** Initialize variables
mID = SPACE(5)
mNAME = SPACE(20)
mADDRESS = SPACE(30)
mTEL = SPACE(7)
mBALANCE = 0.00


currec = RECNO()
GOTO BOTTOM

DO Gets_proc

IF .NOT.changed
  IF .NOT.EOF()          && File wasn't empty
    GOTO currec          && Goto original record, if any.
  ENDIF
  RETURN
ENDIF

* append into DBF fields
APPEND BLANK
REPLACE ID WITH mID
REPLACE NAME WITH mNAME
REPLACE ADDRESS WITH mADDRESS
REPLACE TEL WITH mTEL
REPLACE BALANCE WITH mBALANCE
DO disprec
RETURN



************************************************
* Gets_proc: @..GET's for addrec & editrec
************************************************
PROC Gets_proc
PRIVATE rkey, result


editing = .T.                && Tell disprec we're editing
*** get input fields into memvar duplicates
SET COLOR TO ,N/W
@ 10, 26 GET mID
@ 12, 25 GET mNAME
@ 14, 27 GET mADDRESS
@ 16, 29 GET mTEL
@ 18, 27 GET mBALANCE PICTURE '9999999.99'
```

```
READ

rkey = READKEY()
IF rkey > 256
  rkey = rkey - 256
ENDIF


changed = rkey <> 12    && Escape
RETURN


*********************************************************************
* disprec: displays current record, along with recno()
* and deleted().
* This procedure should only be used to display fields while
* not appending.  If you do call this when the user is adding a new
* record, this will display the last record displayed, not the new
* record (since it hasn't been written yet).
*********************************************************************
PROC disprec

IF EOF()
  RETURN          && Avoid empty-file problems
ENDIF

SET COLOR TO N/W
@ 10, 26 SAY ID
@ 12, 25 SAY NAME
@ 14, 27 SAY ADDRESS
@ 16, 29 SAY TEL
@ 18, 27 SAY BALANCE PICTURE '9999999.99'

RETURN


*********************************************************************
* dispmrec: displays memvars for current record.
* This procedure should only be used to display fields while
* editing.  It displays memvar versions of fields.
*********************************************************************
PROC dispmrec

SET COLOR TO N/W
@ 10, 26 SAY mID
@ 12, 25 SAY mNAME
@ 14, 27 SAY mADDRESS
@ 16, 29 SAY mTEL
@ 18, 27 SAY mBALANCE PICTURE '9999999.99'

RETURN
```

■

ส่งท้าย

โปรแกรมที่สร้างขึ้นนี้สามารถนำไปใช้ในดีเบสได้ทันที นอกจากนั้น UI ยังสามารถเชื่อมโยงชิ้นงานต่าง ๆ สามารถสร้างเมนูที่สวยงามตามต้องการ สำหรับในตอนต่อไปจะอธิบายถึงรายละเอียดการใช้เพิ่มเติม

สำหรับผู้สนใจที่ต้องการซื้อโปรแกรม UI ไว้ใช้งานของตนเองติดต่อได้ที่ร้านบุ๊คเซสท์ สยามสแควร์ ซอย 2 โทร. 251-6334